

Deep Learning Techniken als Boundary Objects zwischen Entwicklungs- und Anwendungsfeld

Philip Roth, Matthias Dorgeist, Astrid Schulz

1. Einleitung

Personen aus ganz unterschiedlichen Abteilungen, Disziplinen oder Hierarchieebenen beteiligen sich oftmals an Projekten zur Entwicklung neuer Technologien. Eine Herausforderung in solchen Projekten ist es daher, die heterogenen Wissensbestände aus unterschiedlichen Feldern produktiv zu integrieren (Hahn 2013; Heidenreich/Mattes 2017). Praxistheoretische Perspektiven legen dabei nahe, dass sich die einzelnen Wissensbestände nicht auf einfache Weise integrieren lassen, da es schwierig ist, diese zu reflektieren und zu kommunizieren. Das spezifische Wissen der Akteur:innen (zum Beispiel von Ingenieur:innen, Informatiker:innen oder Ökonom:innen) lagert sich in ihren alltäglichen Praktiken und spezifischen Praxiszusammenhängen ab, ist damit Teil ihres Habitus und des jeweiligen Feldzusammenhangs und ist zunächst mehr oder weniger unbewusst. Um die Verständigung und Koordination zwischen den heterogenen Partner:innen in interdisziplinären Projekten zu schaffen, ist es dementsprechend erforderlich, dass die Partner:innen aus ihrem habituell gewohnten Praxiszusammenhang heraustreten und einen Kontextwechsel vollziehen, um einen gemeinsamen Rahmen der Verständigung zu finden (Ibert/Müller/Stein 2014; Heidenreich/Mattes 2017; Carlile 2002). In der Forschung zur Verständigung heterogener Partner:innen in Entwicklungsprojekten wurde gezeigt, dass Entwicklungsgegenstände – also die jeweils zu entwickelnden Geräte, Programme etc. – selbst als *Boundary Objects* Verständigungs- und Koordinationsprozesse strukturieren können (Leonardi/Bailey/Pierce 2019; Barrett/Oborn 2010; Hahn 2013; Carlile 2002). In diesem Beitrag fragen wir danach, wie Deep Learning Techniken (DLT) als

Entwicklungsgegenstand das Verhältnis von DLT-Entwickelnden und DLT-Anwendenden strukturieren.

Bei der Beantwortung dieser Frage stützen wir uns auf feld- und praxistheoretische Konzepte und nicht, wie die übrigen Beiträge des Sammelbandes, auf das Theorieangebot von Harrison White. Diesen Zugang haben wir gewählt, da uns die Exploration des empirischen Materials zu der oben genannten Forschungsfrage und einer Fokussierung von Akteursperspektiven und -praktiken geführt hat. Ausgehend davon haben sich für uns vor allem praxistheoretische Vorarbeiten als unmittelbar hilfreich erwiesen. Dementsprechend war es uns mit geringerem Übersetzungsaufwand möglich, den empirischen Gegenstand praxistheoretisch zu fassen und an den Stand der Forschung anzuschließen. Im abschließenden Abschnitt des Beitrags werden wir vor dem Hintergrund unsere Befunde diskutieren, welche weiterführenden Erkenntnisse eine Integration von Whites Theorie in diesem Forschungsfeld verspricht.

In Entwicklungsprojekten kommt es regelmäßig zu Differenzen zwischen den Perspektiven der technischen Entwicklung und den Perspektiven der Anwendung (Ibert/Müller/Stein 2014; Heidenreich/Mattes 2017; Degele 1996). Als Phasen eines Entwicklungsprozesses folgen sie unterschiedlichen Logiken, was im Prozess zu Schwierigkeiten führen kann, indem beispielsweise in der technischen Entwicklung Anwendungsaspekte vernachlässigt werden, die zu »zum Scheitern verurteilten« Prototypen werden, es zu keiner ausreichenden Zusammenarbeit kommt oder Anwendende nicht realisierbare technische Anforderungen formulieren (Degele 1996). Dem begegnend werden seit den 1980er Jahren mögliche Anwendende gezielt in Produktentwicklungsprojekte eingebunden, um ihre Kenntnisse frühzeitig fruchtbar zu machen (Hippel 1986; Bogers, et al. 2017).¹ Mit dieser Verschiebung werden Anwendende selbst zu Beteiligten im Entwicklungsprozess, indem sie ihr Handlungswissen in den Entwicklungsprozess einbringen. Doch mit der organisationalen Einbindung der Anwendenden in Entwicklungsprojekte ist noch nicht zwingend eine Überbrückung der Differenzen gelungen. So spiegelt sich das unterschiedliche Interesse weiterhin in einer unterschiedlichen Problematisierung der Bedeutung eines spezifischen Entwicklungsobjektes und der Befolgung verschiedener Handlungslogiken wider. Deshalb werden unter Rückgriff auf

1 Siehe auch Degele (1996: 65) zur Entwicklung der Partizipativen Systementwicklung, die beginnt, das Gelingen von Entwicklungsprojekten im Softwarebereich in nicht-technischen Faktoren zwischen Entwickler:innen und Anwender:innen zu verorten.

Praxistheorien Anwendende und technische Entwickler:innen in der diesbezüglichen Forschung als Akteur:innen konzipiert, die in unterschiedliche *Soziale Felder* eingebettet sind (Levina/Vaast 2005, 2008; Bjerregaard 2010). Soziale Felder entstehen, wenn Akteur:innen praktisches Wissen miteinander teilen und diesem folgend nach den gleichen Regeln kontextspezifisch kompetent handeln (Bourdieu/Wacquant 2006; Levina/Vaast 2005). Grenzen zwischen Feldern bestehen dementsprechend dort, wo Diskontinuitäten und Brüche zwischen diesen praktischen Eigenlogiken vorzufinden sind (Levina/Vaast 2005, 2008). Die Brüche müssen dabei nicht offensichtlich im Alltag zu Tage kommen, sondern können sich zum Beispiel auch in Formen des aneinander Vorbeiredens andeuten. Entscheidend für eine gelungene Integration beider Felder ist daher, das eigene praktische Handlungswissen für Akteur:innen des anderen Feldes verständlich machen zu können und so zu einer Überwindung der Feldgrenzen zu gelangen (Leonardi/Bailey/Pierce 2019; Heidenreich/Mattes 2017; Levina/Vaast 2005).

In der Theorie des Boundary Objects spielen bei der Koordination und Verständigung zwischen den heterogenen Partner:innen innerhalb solcher Projekte neben menschlichen Akteur:innen auch die technischen Entwicklungsobjekte² eine wichtige Rolle. Diese haben sich als wichtiger Faktor bei der Vermittlung zwischen den Feldern erwiesen (Barrett/Oborn 2010; Levina/Vaast 2005; Leonardi/Bailey/Pierce 2019). Boundary Objects sind abstrakte oder konkrete Gegenstände, auf die sich in verschiedenen Sozialen Feldern

-
- 2 Der Entwicklungsgegenstand erfährt in verschiedenen Phasen des Entwicklungsprozesses unterschiedliche Repräsentationen (z.B. Skizzen, Modelle, Prototypen). Beispielsweise ermöglichen Prototypen, die zwischen technisch Entwickelnden und Anwendenden zur Diskussion gestellt werden können, eine Verständigung zwischen den beteiligten Akteur:innen. Außerdem sind die Repräsentationen der Entwicklungsgegenstände in verschiedenen Entwicklungsphasen mit spezifischen, darauf bezogenen Praktiken assoziiert, die sich sowohl aus den technischen Anforderungen des Entwicklungsgegenstandes in der jeweiligen Phase als auch aus den darauf bezogenen sozialen Konventionen ergeben. Als Boundary Objects schließen Entwicklungsgegenstände dementsprechend die spezifischen Entwicklungsarchitekturen ein, die die auf das jeweilige Endprodukt hin ausgerichteten Zwischenergebnisse und Praktiken beschreiben. Was konkret entwickelt wird, kann sich stark voneinander unterscheiden. Damit gehen unterschiedliche Typen von Boundary Objects einher. Infolgedessen unterscheidet sich auch systematisch, wie Anwendende und technisch Entwickelnde sich in die Entwicklungsprozesse einbringen können und wie die Entwicklungsprozesse verlaufen.

verortete Akteur:innen beziehen. Sie ermöglichen wechselseitige Verständigung und Koordination unterschiedlicher Partner:innen (Star/Griesemer 1989). Dabei zeigt die bisherige Forschung aber auch, dass diese nicht nur Verständnis füreinander ermöglichen, sondern den Entwicklungsprozess diesem entgegen strukturieren können und die Überwindung der Feldgrenzen blockieren. Dies geschieht zum Beispiel, indem sie Akteur:innen aus unterschiedlichen Feldern ungleichen Einfluss ermöglichen und spezifische Verständigungsergebnisse nahelegen (Barrett/Oborn 2010; Leonardi/Bailey/Pierce 2019). Diese Perspektive unterstützt dabei, die Interaktion der Felder über den Entwicklungsgegenstand zu fokussieren (Leonardi/Bailey/Pierce 2019; Hahn 2013: 245f.). Als gelungene Form der Verständigung gilt dabei ein Prozess, der es den jeweiligen Partner:innen ermöglicht, ihr Wissen in die Gestaltung für alle verständlich einbringen zu können. Die so gewonnene Diversität der Zugänge in der Entwicklung reduziert im Ideal die Kontingenz späterer Anwendungsfälle.

In diesem Beitrag argumentieren wir, dass sich Deep Learning Techniken (DLT) als Boundary Objects systematisch von anderen Entwicklungsgegenständen unterscheiden und die Verständigung und Koordination zwischen DLT-Anwendenden und DLT-Entwickler:innen – im Folgenden Programmierende genannt – auf spezifische Weise strukturieren. Wir werden zunächst DLT, Soziale Felder und Boundary Objects einführen, um unser Argument anhand des Forschungsstandes zu entwickeln. Anschließend werden wir das grundsätzliche Argument an empirischem Material konkretisieren. Hierfür ziehen wir zwei Fallbeispiele heran, deren Grundlage Interviews mit Beteiligten solcher Projekte sind, die inhaltsanalytisch ausgewertet wurden. Im Zentrum stehen dabei die DLT-Entwicklungsprozesse aus zwei Projekten, in denen *supervised* DLT zur Bildanalyse in den Lebenswissenschaften entwickelt werden. Anhand dieser Daten zeigen wir, dass DLT als Boundary Objects zu einer paradoxen Einbindung von Lebenswissenschaftler:innen (als DLT-Anwendende) führen: Einerseits erfordert die Entwicklung von DLT eine starke Einbindung der Akteur:innen aus dem Anwendungsfeld, da die Programmierenden (technische Entwickler:innen) von Anfang an darauf angewiesen sind, dass Akteur:innen aus dem Anwendungsfeld ihr Wissen in Form von Lerndaten zur Verfügung stellen und diese von ihnen so aufbereitet werden, dass der Algorithmus von ihnen lernen kann. Ohne adäquate Lerndaten der Lebenswissenschaftler:innen ist es kaum möglich, den DLT-Entwicklungsprozess

erfolgreich abzuschließen.³ DLT als Boundary Objects fördern dementsprechend eine konsequente und frühe Einbindung der Akteur:innen aus dem Anwendungsfeld in den Entwicklungsprozess. Zugleich legt diese Einbindung andererseits aber eine selektive Form der Integration nahe. Innerhalb der Entwicklungsprojekte zeigen sich Tendenzen, die Angewiesenheit auf das Wissen aus dem Anwendungsfeld zu reduzieren und so den Verständigungsprozess zu umgehen. Die DLT ersetzen weitestgehend das Verstehen der Perspektive der Anwendenden (vermittelt durch die von diesen bereitgestellten Lerndaten). Dies reduziert die Relevanz für Programmierende, sich intensiv mit den Anwendenden und den alltäglichen Praktiken des Anwendungsfeldes zu befassen. Diese stark fokussierte und selektive Einbindung der Anwendenden in DLT-Entwicklungsprojekte folgt der Logik einer Generalisierung und verdeutlicht eine Relevanzminderung der alltäglichen Anforderungen der Anwendungsfelder.

Der Artikel leistet damit insbesondere zweierlei. Zunächst trägt er zu einem besseren Verständnis der Spezifika von DLT-Entwicklungsprozessen bei, auf deren Grundlage Szenarien zur Entwicklung künstlicher Intelligenz, die auf der Analyse der technischen Möglichkeiten beruhen, sinnvoll ergänzt werden können. Weiterhin schließt der Artikel an die kritische Forschung zu Boundary Objects an. Entgegen der Annahme, Boundary Objects wirkten grundsätzlich verbindend und egalisierend, weist dieser Beitrag darauf hin, dass Boundary Objects auch Ungleichheiten schaffen oder verfestigen können (Barrett/Oborn 2010; Hawkins/Pye/Correia 2017; Levina/Vaast 2005). Daran anschließend zeigen wir, dass Boundary Objects nicht nur dabei helfen können, Grenzen zu überwinden, sondern es ebenso möglich ist, dass sie das Aufrechterhalten dieser Grenzen fördern.

2. Deep Learning Techniken als Boundary Objects zwischen Anwendungs- und Entwicklungsfeld

Im Folgenden werden wir unser Argument entwickeln, dass DLT als Boundary Objects das Verhältnis von Anwendenden und Programmierenden in DLT-Entwicklungsprojekten systematisch strukturiert und damit die empirische Un-

3 Dabei ist anzumerken, dass wir uns auf supervised Learner konzentrieren, denen die Muster, die sie erkennen sollen, beigebracht werden und diese somit beim Lernen überwacht werden.

tersuchung vorbereiten, die genauer herausstellen soll, in welcher Weise dies geschieht. Dazu werden wir in diesem Abschnitt die drei für unseren Beitrag zentralen Konzepte *Deep Learning Techniken*, *Soziale Felder* und *Boundary Objects* einführen und auf unseren Anwendungsfall beziehen.

Deep Learning Techniken

In den Lebenswissenschaften kommen immer häufiger Deep Learning Techniken (DLT) zum Einsatz. Sie entwickeln diese in gemeinsamen Projekten mit Programmierenden, um aus großen Datenmengen Muster abzuleiten und diese Mustererkennung für weitere Datensätze zu automatisieren. Fortschritte wurden in den letzten Jahren besonders im Bereich der Bildverarbeitung erlangt. Inzwischen stehen riesige Mengen an Bilddaten für das Training bereit und moderne und leistungsstarke Computer sind in der Lage, diese zu verarbeiten, sodass DLT in spezifischen Aufgabenfeldern mittlerweile bessere oder ähnlich verlässliche Ergebnisse wie Menschen hervorbringen (Hosny et al. 2018). Machine Learning produzierte in den vergangenen zehn Jahren besonders dann Schlagzeilen, wenn die besten menschlichen Spieler:innen zum Beispiel im Spiel »Go« von Programmen geschlagen wurden (Serre 2019). Doch neben diesen – die Aufmerksamkeit bündelnden – Ereignissen verbreitet sich die Technik auch zunehmend im Alltag kleinerer Forschungs- und Entwicklungsprojekte. Dies liegt daran, dass DLT heutzutage nach einem Baukastenprinzip und auf der Grundlage vorgefertigter Programmzeilen programmiert werden kann und die technische Hardware für jede:n verfügbar ist. Die Medizin, bzw. weiter gefasst die Lebenswissenschaften, sind der Anwendungsbereich, in dem die Einbindung der DLT in verschiedene Projekte am weitesten fortgeschritten ist (Buchkremer/Witte/Matusiewicz 2020: 388). Dies spiegelt sich auch in dem großen Investitionsaufkommen wider (Razzak/Naz/Zaib 2018: 335). In den Lebenswissenschaften können DLT zum Beispiel wie in den von uns ausgewählten Beispielen unterstützend zur Detektion von Aneurysmen in der Radiologie oder zur Beobachtung von Zellwachstumsprozessen in der Biologie eingesetzt werden.

Trotz der zurückliegenden Entwicklungssprünge ist die alltägliche und vor allem medial gestützte Wahrnehmung von Künstlicher Intelligenz im Sinne eines Hypes zu verstehen (Aust 2021). Überbordende Erwartungen und Ängste gegenüber der Technologie, vermittelt über das Narrativ einer Ersetzung des Menschen durch die Technik wie am Beispiel »Go«, geraten allzu schnell in die Gefahr, dem tatsächlichen Entwicklungsstand nicht gerecht zu werden. Denn

Probleme, wie die begrenzte Fähigkeit, abstrakte Repräsentationen zu bilden und über die Trainingsdaten hinaus zu generalisieren, sind längst in Fachkreisen bekannt (Serre 2019). Ein Blick in aktuelle Entwicklungsprozesse mag so als ein Beitrag zur Korrektur oder Einordnung dieser Erwartung dienen. In Anbetracht des Entwicklungsstandes von DLT und im Kontrast zur bisherigen DLT-Forschung erscheint es vielversprechend, einen Schritt zurückzutreten und statt der vielfach angestellten Überlegungen zu den Folgen und Risiken bei der möglichen Implementierung von DLT in verschiedene Anwendungsbereiche zunächst DLT-Entwicklungsprozesse genauer in den Blick zu nehmen. Hier stellt sich insbesondere die Frage, wie DLT aufgrund ihrer spezifischen soziotechnischen Qualitäten die Entwicklungsprozesse und das Verhältnis der beteiligten Akteur:innen auf charakteristische Art und Weise strukturieren. Welche Rolle spielen DLT in solchen interdisziplinären Forschungsprojekten? Welche Probleme zeigen sich in den Entwicklungsprozessen? Erfordern DLT aufgrund ihres spezifischen Charakters – scheinbar leicht in der Anwendung, aber höchst komplex in ihrer Grundstruktur – besondere Verständigungsformen innerhalb der Entwicklungsteams? Was bedeutet das für die weitere Diffusion dieser Innovation?

Damit entwickeln wir an dieser Stelle eine Perspektive, die DLT in ihrer Gesamtheit als Technologie zu begreifen. Die Aufbereitung, Verarbeitung und Ausgabe der Daten prägen im Entwicklungsprozess ein ganzes Set unterschiedlicher Programme und informationstechnische Architekturen. Gleichzeitig ist unser Fokus weniger darauf gerichtet, wie die Technologie arbeitet, sondern wie mit ihr gearbeitet, geforscht und wie sie entwickelt wird.

Soziale Felder

DLT werden wie andere avancierte Techniken heute typischerweise partizipativ entwickelt und Anwendende von Anfang an eingebunden. Die Forschung zur partizipativen Technikentwicklung hat gezeigt, dass die Konstellationen der verschiedenen Akteur:innen, die an diesem Prozess beteiligt sind, herausfordernde soziotechnische Strukturen sind, die spezifische Dynamiken, Interaktionen und Diskontinuitäten bei der Vermittlung heterogener Wissensbestände produzieren. In unserem Beitrag geht es uns vor diesem Hintergrund darum, zu verstehen, wie diese Konstellationen im Fall von DLT-Entwicklungsprozessen konstituiert werden. Insbesondere werden wir dabei danach fragen, welche Rolle die spezifischen Qualitäten von DLT bei der

Formation der interdisziplinären Akteurskonstellationen spielen. So zeigen sich beispielsweise konkret beim »supervised learning« Herausforderungen in der Phase der Annotation der Lerndaten, bei der Expert:innen des Anwendungsfeldes ihr Wissen integrieren müssen, während bei der Optimierung und Anpassung des Learners Programmierende die Führung übernehmen.

In der Forschung zu Entwicklungsprojekten, in denen Anwendende temporär mit Programmierenden zusammenarbeiten, hat sich das Konzept des Sozialen Feldes als besonders fruchtbar erwiesen (Levina/Vaast 2005; Leonardi/Bailey/Pierce 2019). Für das Konzept sozialer Felder sind vor allem zwei Aspekte charakteristisch (Martin 2003; Fligstein/McAdam 2011; Roth 2019): erstens die inhärente Logik der Praxis in Feldern und zweitens die relative Position der Akteur:innen. Beide Aspekte sind eng miteinander verbunden. Die inhärente Logik von Feldern legt im Wesentlichen die Regeln fest, nach denen Akteur:innen in einem Feld soziale Anerkennung (symbolisches Kapital) und andere Formen von Kapital akkumulieren können, die innerhalb (und außerhalb) des Feldes als erstrebenswert definiert werden und Gestaltungsmöglichkeiten konstituieren. Die hierarchischen Positionen sind dementsprechend eine wichtige Triebkraft für die alltägliche Reproduktion der feldspezifischen Logiken.

Felder und ihre inhärenten praktischen Logiken werden produziert, reproduziert und verändert, indem Akteur:innen immer wieder in interaktive Praktiken miteinander eintreten und dabei Praxiswissen generieren und teilen (Berends/Sydow 2019; Barley/Tolbert 1997; Bourdieu 2009: 174; Giddens 1997: 52). Das geteilte Wissen konstituiert ähnliche Habitus und Institutionen (Barley/Tolbert 1997; Hallett 2003). Habitus werden als geteilte Wahrnehmungs-, Bewertungs-, Denk- und Handlungsschemata verstanden (Bourdieu 1993: 93ff; Krais 2004; Lizardo 2004). Akteur:innen, die solche habituellen Schemata teilen, richten ihre Handlungen routinemäßig an denselben impliziten und expliziten Regeln und Zielen aus. Darüber hinaus ermöglichen ihnen gemeinsame Schemata, zuverlässige Annahmen darüber zu treffen, wie andere ihre Handlungen und Äußerungen interpretieren und bewerten werden. Diese gemeinsamen Erwartungen definieren Institutionen (Battilana 2006; Berger/Luckmann 1980; Friedland/Alford 1991). In dem Maße, in dem Akteur:innen nach erfolgreichem Verständnis, Anerkennung, sozialem Status und damit verbundenen Ressourcen im Feld streben und umgekehrt soziale Sanktionen vermeiden, orientieren sie ihr Handeln nicht nur routinemäßig, sondern auch strategisch an den impliziten und expliziten Regeln des Feldes (Friedland/Alford 1991; Bourdieu/Wacquant 2006: 129ff; Fligstein/McAdam

2011). Grenzen zwischen Feldern entstehen dort, wo sich das auf vergleichbare Praktiken bezogene Wissen – und die darin begründeten Institutionen und habituellen Dispositionen – voneinander unterscheiden (Levina/Vaast 2005, 2008; Bourdieu/Wacquant 2006; Roth 2019).

Das Feldkonzept ist besonders prominent auf gesellschaftliche Teilbereiche wie Wissenschaft und Wirtschaft angewandt worden. Weiterhin wurde das Konzept aber auch für die Analyse subtilerer Unterschiede zwischen kleineren Kollektiven fruchtbar gemacht (Roth 2019; Levina/Vaast 2005). Insbesondere zur Analyse von Entwicklungsprojekten, in denen Akteur:innen mit heterogenem praktischem Wissen temporär kollaborieren sollen, hat sich das Konzept als fruchtbar erwiesen (Carlile 2002; Majchrzak/Jarvenpaa/Bagherzadeh 2015; Berends/Sydow 2019). In interdisziplinären Kollaborationsprojekten, Partnerschaften zwischen Wissenschaft und Wirtschaft oder bei der Einbindung von Anwendenden in Entwicklungsprozesse, geht es in der Regel darum, die heterogenen Perspektiven und Kenntnisse der unterschiedlichen Akteur:innen erfolgreich zu kombinieren, um Innovationen zu schaffen. Das Feldkonzept weist jedoch darauf hin, dass die Differenzen im Wissen der Akteur:innen oftmals tiefgreifender sind als das auf den Innovationsgegenstand bezogene Faktenwissen. Diesbezügliche Unterschiede können für Kollaborationen grundlegende Praktiken wie die Identifikation der im Entwicklungsprozess auftretenden Probleme, der erstrebenswerten Ziele, die Organisation von Zusammenarbeit oder die Kommunikation betreffen. Erschwerend kommt weiterhin hinzu, dass dieses feldspezifische Wissen von den Akteur:innen in hohem Maße implizit angewandt und als selbstverständlich vorausgesetzt wird. Zwischen Kooperationspartner:innen aus unterschiedlichen sozialen Feldern ergeben sich daher oftmals Interessens-, Koordinations- und Verständigungsschwierigkeiten (Carlile 2002; Gassol 2007; Leonardi/Bailey/Pierce 2019). Die unterschiedlichen Wissensbestände, die oft als Chance für Innovationen und wechselseitiges Lernen gesehen, propagiert und befördert werden, sind dementsprechend zugleich eine Barriere für erfolgreiche Zusammenarbeit.

Voraussetzung dafür, dass diese Barrieren überwunden und Kooperationen über Feldgrenzen hinweg erfolgreich umgesetzt werden können, ist, dass es gelingt, die Unterschiede zwischen den Feldern temporär zu überwinden, indem relevantes Praxiswissen geteilt, generiert, habitualisiert und institutionalisiert wird und es zu gemeinsamen Praktiken zwischen Akteur:innen aus den beiden beteiligten Feldern kommt. Erfolg heißt hierbei aber nicht nur, dass ein spezifisches Produkt entwickelt wird, sondern er misst sich

auch an dem Grad der wechselseitigen Verständigung. Nach dem Konzept der Sozialen Felder gelingt die Verständigung nur innerhalb eines Feldes, sodass Entwicklungsprozesse an der Konstitution eines gemeinsamen Verständnisses erst arbeiten müssen. Die spezifischen kulturellen Logiken ergeben sich im Detail aus den näheren Praxiszusammenhängen der Akteur:innen, womit die kulturelle Differenzierung ausgehend von den makroskopischen Feldern wie Politik, Wirtschaft oder Wissenschaft auf kleiner geartete Interaktionskonstellationen bzw. Arbeitsgemeinschaften herunterzuskalieren ist.⁴ In den untersuchten Projekten sind es die individuellen Forschungskontexte, mit der spezifischen Fachcommunity, den Konferenzen und Fachzeitschriften, die dem jeweiligen wissenschaftlichen Streben die strategische Logik geben.

Bezogen auf DLT-Entwicklungsprojekte sind die Felder das *Entwicklungsfeld*, indem DLT-Programmierende DLT entwickeln, sich mit anderen Programmierenden in diesem Feld austauschen und sich ihnen gegenüber profilieren und das *Anwendungsfeld*, indem etwa Ärzt:innen alltäglich Patient:innen Diagnosen stellen, sie untersuchen und behandeln oder in denen Biolog:innen Bildmaterial analysieren und sich darüber mit ihren Kolleg:innen austauschen. Die Bereiche verfügen über spezifische disziplinäre Kenntnisse, die für die Entwicklung von DLT-Anwendungen für bildgebende Verfahren in der Medizin und der Biologie entscheidend sind.

An den hier fokussierten Entwicklungsprojekten von DLT für die Lebenswissenschaften sind typischerweise Akteur:innen aus der Informatik und dem jeweiligen lebenswissenschaftlichen potentiellen Anwendungsbereich beteiligt. In der bisherigen Forschung zu Softwareentwicklung hat sich immer wieder gezeigt, dass sich die sozial geteilten Annahmen über Funktion, Anwendung und Nutzer:innen der DLT systematisch voneinander unterscheiden und deren Ausgestaltung auf unterschiedliche Weise strukturieren können (Degele 1996; Amershi et al. 2019). Typischerweise fokussieren Programmierende die effiziente sowie in sich schlüssige Gestaltung der Software und blenden die spezifische Anwendungssituation und die sich daraus ergebenden Bedarfe weitestgehend aus. Sie werden nur am Rande und insofern berücksichtigt, als Programmierende von ihrer Nutzungssituation auf die

4 Disziplinen sind hierbei, wie Schmitt (2019: 65) schreibt, ungenau, wenn es darum geht die kulturellen Unterschiede in der jeweiligen Forschungspraxis von Wissenschaftler:innen zu beschreiben. Möglich wäre es auch sich hier über die Begriffe der epistemischen Kulturen (Knorr Cetina 2002) oder den wissenschaftlichen Produktionsgemeinschaften (Gläser 2006) zu nähern.

Situation der zukünftig Nutzenden schließen. Solche Schlüsse zeichnen sich in der Regel durch eine rationalistische, stark an Stereotypen ausgerichtete und im Hinblick auf die zentrale Funktion der Anwendung reduktionistische Konzeption der Anwendungsszenarien und Interessen der Anwendenden aus. Dominieren Programmierende also die DLT-Entwicklungsprozesse, erhöht dies das Risiko, dass DLT entwickelt werden, die in der Anwendung nicht praxistauglich sind und somit die Entwicklungsprojekte gegebenenfalls scheitern. Gelingt es doch, die stark durch die Programmierenden geprägten DLT in der Praxis zu etablieren, sind dagegen stärkere Diskontinuitäten in den Anwendungsfeldern zu erwarten. Dabei werden die in die DLT eingeschriebenen Annahmen der Programmierenden tendenziell praktisch reproduziert, weil Anwendende bei der Nutzung dazu veranlasst werden, diesen Erwartungen zu entsprechen. Ziele und Praktiken der Anwendenden geraten dabei gegenüber dem Bestreben der Programmierenden, die DLT in sich zu optimieren, in den Hintergrund.

Für die Ausgestaltung der DLT-Entwicklungsprojekte und ihrer nachhaltigen Folgen ist es entscheidend, inwiefern das Entwicklungs- und Anwendungsfeld durch die Projekte punktuell und temporär integriert werden (Levina/Vaast 2008; Leonardi/Bailey/Pierce 2019). Ausschlaggebend dafür ist das konkrete Verhältnis, das Anwendende und Programmierende in DLT-Entwicklungsprojekten zueinander einnehmen und ob eine Verständigung angestrebt wird und gelingt. Definiert wird dies besonders dadurch, inwiefern Akteur:innen gemeinsam verständigungsorientiert aufeinander Bezug nehmen oder ob versucht wird, feldspezifische Eigenlogiken gegeneinander durchzusetzen. Wir fragen dementsprechend im Folgenden danach, wie Anwendende und Programmierende sich und die Eigenlogiken ihrer Felder im Rahmen von DLT-Entwicklungsprojekten in ein Verhältnis zueinander stellen.

Boundary Objects

In der Forschung zu interdisziplinären Entwicklungsprojekten hat sich gezeigt, dass Boundary Objects das Lernen und die Relationen zwischen den Feldern entscheidend strukturieren können. Wir werden im folgenden Deep Learning Techniken selbst als Boundary Objects konzipieren und danach fragen, wie diese spezifische Technologie die Relationen zwischen Anwendenden und Programmierenden strukturiert. Als Boundary Objects definieren Star und Griesemer abstrakte oder konkrete Gegenstände, »which are both plastic

enough to adapt to local needs and the constraints of the several parties employing them, yet robust enough to maintain a common identity across sites.« (1989: 393)

Die dem Ansatz zugrunde liegende Annahme ist, dass, indem sich Akteur:innen mit verschiedenen Kenntnissen und Perspektiven gemeinsam auf dieselben Boundary Objects beziehen, diese durch geteilte Bezüge eine bessere Verständigung ermöglichen können. Zugleich können Boundary Objects wirksam werden, indem sie den Einfluss verschiedener Akteur:innen auf den Entwicklungsprozess verstärken oder reduzieren.

In Entwicklungsprojekten kommen Prototypen der zu entwickelnden Artefakte eine zentrale Rolle dabei zu, die Explikation impliziten Anwenderwissens zu unterstützen und somit eine Verständigung zwischen Programmierenden und möglichen Anwendenden zu fördern. Beispielsweise explizieren Programmierende selbstverständliche Annahmen, indem sie ihre Programme an Prototypen ausrichten. Mögliche Anwendende können sich dann, wenn sie das Programm in die Anwendung übertragen sehen, an den entsprechenden Ausgestaltungen stoßen. Anhand der entsprechenden Reibungspunkte lassen sich Differenzen zwischen den impliziten Annahmen identifizieren, die deren Artikulation befördern. So können die gemeinsamen Bezüge auf Boundary Objects ein besseres Verständnis für die Perspektiven der anderen schaffen (Leonardi/Bailey/Pierce 2019). Neben physischen Prototypen (Bechky 2003; Carlile 2002) werden in der Literatur beispielsweise auch Skizzen (Bødker 1998), Anwendungsszenarien (Bødker 2000) oder Formulare (Star/Griesemer 1989) als Boundary Objects identifiziert. Entscheidend ist jeweils weniger die Art der Objekte, sondern vielmehr, dass sie zur interaktiven Verständigung zwischen unterschiedlichen Partner:innen beitragen (Levina/Vaast 2005; Barrett/Oborn 2010).

Wie oben bereits erwähnt, werden Anwendende heute typischerweise in (DLT-)Entwicklungsprozesse partizipativ eingebunden. Eine erhebliche Schwierigkeit besteht in interdisziplinären Projekten insofern, als die Programmierenden und Anwendenden ihre praktischen Bedarfe nicht ohne weiteres artikulieren können. Das Wissen über ihre Praxis ist in aller Regel in hohem Maße implizit und schließt zahlreiche unhinterfragte Annahmen ein. Um das Wissen zu erschließen und in den Entwicklungsprozess einzubeziehen, bedarf es daher einer systematischen Explikation des feldspezifischen Wissens, die sich kaum alleine sprachlich realisieren lässt (Roth/Diefenbach 2021). Entscheidend dafür, ob das implizite Anwender:innenwissen erschlos-

sen werden kann und in welchem Maße es Niederschlag in der Gestaltung der DLT findet, kann durch das Wirken von Boundary Objects beeinflusst werden.

Untersuchungen, die Boundary Objects in Entwicklungsprozessen betrachten, zeigen weiterhin, dass diese weder an sich eine spezifische Wirkung auf Entwicklungsprozesse und Akteurskonstellationen entfalten, noch, dass Boundary Objects in dem Sinne neutral sind, dass Akteur:innen sie nach Belieben und entsprechend ihrer reflexiven Ziele einsetzen können. Vielmehr entscheidet die spezifische praktische Einbindung von Objekten, ob sie zu Boundary Objects werden und welche spezifische Wirkung sie als solche entfalten (Leonardi/Bailey/Pierce 2019; Levina/Vaast 2005). Wie Boundary Objects wirken, ergibt sich dementsprechend aus dem praktischen Zusammenspiel der bestehenden Akteurskonstellationen und den technischen Artefakten. Beide strukturieren diesen Prozess in gewissem Maße durch ihre Charakteristika.

Die kritische Forschung zeigt daran anschließend auch, dass die Anwendung von Boundary Objects nicht notwendigerweise zu einer wechselseitigen Verständigung und zu einer egalitären Beteiligung von Programmierenden und Anwendenden führen muss. Vielmehr zeigt sich, dass Boundary Objects von Macht durchdrungen sind (Hawkins/Pye/Correia 2017; Barrett/Oborn 2010). Die Gestaltung und das Verstehen von Boundary Objects setzt mitunter spezifische Kenntnisse und Ressourcen voraus. Sind diese Kenntnisse und Ressourcen asymmetrisch zwischen den Akteur:innen verteilt, sind diejenigen, die über die entsprechenden Ressourcen und Kenntnisse verfügen, in höherem Maße dazu in der Lage, ihre Interessen, aber auch ihre unhinterfragten Annahmen im Entwicklungsprojekt durchzusetzen (Barrett/Oborn 2010; Levina/Vaast 2005). Wir verstehen Boundary Objects dementsprechend als Objekte, auf die sich Akteur:innen aus verschiedenen Feldern interaktiv beziehen und gehen davon aus, dass diese Bezüge das Verhältnis der Akteur:innen strukturieren. Boundary Objects können Verständigung und Koordination über Feldgrenzen hinweg dementsprechend symmetrisch befördern. Sie können aber genauso gut asymmetrische Konstellationen hervorbringen, indem sie Verständigung und Koordination selektiv verhindern.

Der Forschung zu Boundary Objects folgend gehen wir davon aus, dass DLT als Boundary Objects im Entwicklungsprozess eine wichtige Rolle bei der Formation von Akteurskonstellationen spielen. Weiterhin nehmen wir an, dass sich die besonderen Qualitäten von DLT in besondere Akteurskonstellationen übersetzen, die über diese Perspektive sichtbar gemacht werden können. Im Folgenden geht es uns darum, diese Zusammenhänge anhand

zweier Fallbeispiele genauer zu beleuchten. Um die spezifischen Verschiebungen innerhalb von Entwicklungsprozessen differenziert nachvollziehen und identifizieren zu können, untersuchen wir die Konstellationen innerhalb und entlang der einzelnen Phasen und Momente der DLT-Entwicklungsprojekte.

3. Methode

Ausgehend von dem Vorhaben eines Arbeitskreises von Mitarbeiter:innen des Lehrstuhls für Technik- und Organisationssoziologie der RWTH Aachen wurden von 2019 bis 2021 Interviews mit Akteur:innen geführt, die sich in ihren Forschungsaktivitäten und Entwicklungsprojekten mit Machine Learning und Deep Learning auseinandersetzen bzw. maßgeblich in entsprechenden Projekten involviert sind oder diese initiieren.

Aus dem Korpus der geführten Interviews ziehen wir drei von den Autor:innen dieses Beitrags geführte Interviews heran. Diese Interviews beziehen zwei Projekte ein, in denen DLT für eine gezielte Anwendung im Bereich der Bildanalyse entwickelt wurden. Diese beiden Projekte werden als exemplarische Fälle herangezogen und im Folgenden näher beleuchtet.

Dieser Beitrag widmet sich somit zwei anwendungsorientierten Entwicklungs- bzw. Forschungsprojekten, in denen DLT programmiert wurden, um Bilddaten aus dem Bereich der Lebenswissenschaften zu analysieren. Die Betrachtung der beiden Projekte setzt sich aus einer Kombination der jeweiligen Einzelfallanalysen und einer vergleichenden Fallanalyse zusammen. Die Auswahl der beiden Fälle sowie das explorative analytische Vorgehen zielen sowohl darauf ab, etwaige spezifische Gemeinsamkeiten in den Konstellationen und Positionierungen beleuchten zu können, als auch darauf, herausstechende Unterschiede innerhalb solcher anwendungsnahen interdisziplinären DLT-Projekte identifizieren zu können (Yin 2003; Borchardt/Göthlich 2007).

Grundlage der Fallstudien sind drei persönlich geführte problemzentrierte, teilleitfadengestützte Interviews mit Akteur:innen von DLT-Entwicklungsprojekten, die 2019 und 2020 geführt wurden. Die drei Interviewpartner:innen zeichnen sich dadurch aus, dass sie sowohl über Expertise des jeweiligen Anwendungsbereiches als auch über Expertise in der Entwicklung von DLT verfügen. Gleichwohl sie sich klar in einem der Felder verorten lassen, erwarten wir, dass sie in den betrachteten und vorangehenden Projekten als *Boundary*

*Spanner*⁵ fungieren und dadurch in besonderem Maße dazu in der Lage sind, die unterschiedlichen Perspektiven, Interessen und Eigenlogiken der involvierten Anwendungs- und Entwicklungsfelder sowie deren Zusammenspiel in den Projekten zu beschreiben. Befragt wurde ein Universitätsprofessor für Radiologie, der ebenso die Leitung einer Klinik innehat (A2). Er vermittelte im Anschluss an das Interview einen in seinen Projekten tätigen Medizingenieur, mit dem ein weiteres Interview stattfand (A1). Des Weiteren wurde ein Interview mit einem Juniorprofessor für biomedizinische Bildverarbeitung geführt, der in dem Interview seine Projektstätigkeiten als Bioinformatiker schildert (B1). Die Interviews waren teilstandardisiert gestaltet und dem explorativen Charakter des Vorhabens entsprechend narrativ ausgerichtet (Borchardt/Göthlich 2007; Lamnek 2008). Die Interviews bieten somit einen besonderen Zugang zu den Erfahrungen und zu den Perspektiven auf die eigene Tätigkeit. Auch die Positionierungen der beteiligten Akteur:innen in Relation auf die anderen Beteiligten – inklusive der DLT – treten hervor.

Die gewählte explorative Gestaltung ermöglicht ausgehend von den dichten Beschreibungen ein tiefergehendes Verständnis der Prozesse, der Entwicklungen und den Verschiebungen innerhalb dieser Forschungs- und Entwicklungsprojekte. Im Besonderen erscheint dies erkenntnisversprechend, um eventuell konstitutive Momente in solchen DLT-Entwicklungsprozessen und charakteristische Akteurskonstellationen, zu erkennen. Trotz der Limitationen dieses Vorgehens versprechen wir uns erste Beschreibungen und spezifische Einblicke zu generieren, die Grundlage für weiterführende Fragestellungen zu den interdependenten soziotechnischen Konstellationen der Akteur:innen in solchen Projekten sein können (Borchardt/Göthlich 2007).

Die Interviews wurden transkribiert und mithilfe von MAXQDA für die inhaltlich strukturierende Inhaltsanalyse nach Kuckartz aufbereitet (Kuckartz 2018). Das Kategoriensystem wurde mittels eines deduktiv-induktiven Vorgehens entwickelt, ergänzt und fortlaufend ausdifferenziert.

5 Boundary Spanner dient als Bezeichnung für Akteur:innen, die den Austausch des fachlichen Wissens unterschiedlicher Felder erleichtern, ermöglichen und die somit bestehende (disziplinäre) Grenzen überbrücken (Levina & Vaast 2005).

4. Fallbeispiele: DLT Entwicklungsprojekte in den Lebenswissenschaften

Im Folgenden werden zunächst die beiden Fallbeispiele beschrieben, um ein Verständnis ihres Entstehungs- und Entwicklungszusammenhangs herzustellen.

Fallbeispiel A: DLT zur Detektion von Aneurysmen in MRT-Bildern

Dieses Entwicklungsprojekt ist ausgehend von der Qualifizierungsarbeit eines Mediziningenieurs (A1) entstanden, der eine DLT zur Detektion von Aneurysmen entwickelt hat. Als Lerndaten konnte er auf vorhandene Patient:innendaten aus einer Klinik für Neuroradiologie zurückgreifen. Hierbei handelt es sich um MRT-Bilder, die in der Klinik in den letzten 10 Jahren angefertigt wurden, bei denen ein oder mehrere Aneurysmen diagnostiziert wurden (Datensatz aus 85 MRT-Aufnahmen mit 115 diagnostizierten Aneurysmen). Die entwickelte DLT wurde so programmiert, dass er den Bereich, in dem ein Aneurysma identifiziert wurde, im Bild markiert. Die Sensitivität der entwickelten DLT lag ähnlich der von Radiolog:innen bei 90 %. A1 arbeitet an diesem DLT-Entwicklungsprojekt zusammen mit Ärzt:innen, die die bereits bestehenden MRT-Diagnosen überprüfen und die Lerndaten annotieren. Die in den geschriebenen Arztbriefen festgehaltenen zum Teil sprachlich unpräzisen Diagnosen werden von praktizierenden Ärzt:innen bewertet, interpretiert, in »diagnostische Score« (A2) transformiert und so aufbereitet, dass sie als Lerndaten für die DLT fungieren können. In den MRT-Bildern werden die Aneurysmen in jedem Bild exakt markiert, indem die betreffenden Bildbereiche umrandet werden. A1 programmiert dann mit diesen annotierten Lerndaten die DLT, anschließend wird die Güte der DLT an einem anderen Datensatz geprüft und die Ergebnisse wieder von Ärzt:innen bewertet. Die Architektur der DLT wird dabei optimiert und angepasst.

Fallbeispiel B: DLT zur Analyse von Mikroskopbildern biologischer Embryonalforschung

In dem Projekt, das als Fallbeispiel B dient, wird eine DLT entwickelt, die 3D-Bilder und -Videos analysiert, die aus den Aufnahmen von hochauflösenden Mikroskopen stammen. Diese werden zur Analyse von embryonalen Entwicklungsprozessen (hauptsächlich von Zebrafischen) eingesetzt und produzieren

umfangreiche Datenmengen. In diesem Projekt wird eine Kombination aus Detektions-, Segmentierungs- und Trackingnetzwerken entwickelt, die die Zellformänderungen in ihrer zeitlichen Entwicklung erkennen und nachvollziehbar machen können. In diesem Projekt treten die Biolog:innen an den befragten Bioinformatiker (B1) heran, kommunizieren ihm die sie interessierenden Strukturen in den Bildern und bitten ihn, eine DLT für die Bildauswertung zu entwickeln. B1 entwickelt in Folge dieser Abstimmung ein Tool, das die Biolog:innen ähnlich wie das Programm »Paint« nutzen können, um die Bilddaten zu annotieren. Hiermit können sie die sie interessierenden Bildbestandteile präzise kennzeichnen, damit diese als Lerndaten für die Programmierung des Algorithmus eingesetzt werden können. Der nächste Schritt im Prozess nach der Programmierung ist die Optimierung der DLT und die Testung der Detektionsgüte und die daraufhin stattfindende Anpassung des Algorithmus. Im nächsten Schritt wird die entwickelte DLT den Biolog:innen überlassen, die sich gegebenenfalls wieder an den Programmierer wenden, wenn die Qualität der Detektion nicht genügt. Daraufhin würde eine nächste Iterationsschleife und Optimierung durchgeführt werden.

5. Soziotechnische Konstellationen in den DLT-Entwicklungsprojekten

Im Folgenden stehen die spezifischen Charakteristiken von DLT-Entwicklungsprojekten im Fokus. Im Zentrum steht dabei die Frage, wie DLT – verstanden als Boundary Objects – die Akteurskonstellationen in den Entwicklungsprojekten strukturieren.

Erste Beobachtung: Programmierer:innen als Auftragnehmer:innen

In beiden Fallbeispielen werden supervised DLT für eine konkrete Anwendung entwickelt, mittels derer Bilddaten aus den Lebenswissenschaften analysiert werden. Betrachtet man die Entstehungszusammenhänge beider Projekte im Hinblick darauf, zeigen sich Gemeinsamkeiten, aber auch Unterschiede.

In Fallbeispiel A initiiert der Medizingenieur A1 das Projekt ausgehend von seiner eigenen Qualifizierungsarbeit. Dabei profitiert er von dem für ihn leicht zugänglichen bereits bestehenden Datenpool. Er spricht den Leiter der Klinik (A2) an, um seine Projektidee zu verfolgen. Er begründet die Wahl dieses Themas zum einen mit dem allgegenwärtigen »Hype« um den Einsatz

künstlicher Intelligenz und zum anderen mit den Versprechungen des zukünftigen Einsatzes von DLT als Diagnosetool zur Optimierung des Klinikalltags von Radiolog:innen. Radiolog:innen, die einer hohen Arbeitsbelastung und folgenschweren Diagnose- und somit Entscheidungssituationen ausgesetzt sind, könnten mit diesem zusätzlichen Tool entlastet und ihre Diagnosegüte verbessert, bzw. das Übersehen von Aneurysmen verhindert werden. Hierfür eignet sich A1 die erforderlichen Kenntnisse zur Programmierung an, da er zuvor noch nicht mit DLT gearbeitet hat. Der Leiter der Klinik (A2) wertet den Nutzen des Projektes für die tatsächliche Praxis in der Klinik im Interview und damit nach Abschluss des Projektes als gering ein. Aufgrund der mangelnden zugrundeliegenden Datenmenge und der daraus hervorgehenden mangelnden Generalisierbarkeit besitze das Projekt keine reelle Chance, als Tool im Klinikalltag implementiert zu werden. Es fügt sich somit nicht in das Set der verschiedenen technischen Hilfsmittel ein, die bereits im ärztlichen Alltag zum Einsatz kommen. Letztlich seien solche kleinen Projekte (leider nur) als »Spaßforschung« zu sehen und Neugierde eine wesentliche Motivation.

Da A1 bereits Mitarbeiter des interdisziplinären Teams der Klinik war, konnte das Projekt auf personeller Ebene niedrigschwellig ermöglicht werden. In Hinblick auf die Technik wird erläutert, dass die Daten aus der Klinik für die Forschung genutzt werden konnten, sodass auch hier die Ressourcen intern verfügbar waren. Daneben wurde die notwendige Hard- und Software angeschafft und die DLT erscheint den Mediziner:innen als einfache technische Grundarchitektur, die in Form von Bausteinen zusammengesetzt werden kann. Die Mediziner:innen betreiben das Entwicklungsprojekt in diesem Fall als »Spaßforschung«, auf die sie nicht angewiesen sind. Letztlich stellt sich das Tool als »nice to have« dar und könnte in ganz bestimmten Fällen das Portfolio weiterer Technologien der Klinik erweitern sowie die Entscheidungen in der Radiologie unterstützen. Über den Zustand als Prototyp kommt die DLT-Anwendung nicht hinaus. Es wird auch klar, dass die Forschung nicht aus medizinischer Notwendigkeit heraus erfolgt. In diesem Fallbeispiel zeigt sich eine enge Verknüpfung von Anwendungsfeld und Programmierenden. Verschiedene Hürden verhindern, das Projekt größer zu skalieren, doch durch die leichte Verfügbarkeit der notwendigen Ressourcen können die Mediziner:innen mit der DLT Versuche durchführen und sich ausprobieren.

Im zweiten Fallbeispiel B gestaltet sich die Anbahnung des Projekts anders. Die Entwicklung der DLT entsteht auf die Initiative der Biolog:innen hin, die dem Problem begegnen, dass moderne Mikroskope immens große Datensätze produzieren, die sie selbst nicht mehr mit »klassischen« Methoden auswerten

können. Die Entwicklung der DLT geschieht gewissermaßen als ›Auftragsarbeit‹, aus der Situation heraus, dass die Datenmenge für sie nicht mehr handhabbar ist und sie darauf angewiesen sind, die Analyse der Bilddaten zu automatisieren und Kooperationen mit Programmierenden anzubahnen.

Die Lebenswissenschaftler:innen sind somit aufgrund der Entwicklung der technologischen Modalitäten dazu aufgefordert, diese durch Automatisierung bearbeit- und handhabbar zu machen. Zugleich mangelt es den Lebenswissenschaftler:innen typischerweise an den dazu erforderlichen Kenntnissen und Kompetenzen. Die entsprechenden Programmierpraktiken entziehen sich daher sowohl ihrer Einsicht als auch ihren Gestaltungsmöglichkeiten. Infolgedessen sind sie auf die technische Expertise von Programmierenden angewiesen. Hierbei bedarf es vor allem informatischer Expertise, da die zu lösenden Probleme vor allem als technische Probleme erscheinen. Die Biolog:innen sind insofern in hohem Maße in ihrer eigenen Disziplin und ihren Möglichkeiten der Problemlösung von der Informatik abhängig. Die Anbahnung des Projekts erscheint dabei jedoch in Form einer Dienstleistung, die erbracht und in Auftrag gegeben wird. Die Entwicklung von DLT scheint sich hier zwischen den Polen von Abhängigkeit und Auftragsarbeit zu bewegen, wodurch die disziplinären Grenzen eher bestehen bleiben, als wenn die Projekte als gemeinsame Forschungs- und Entwicklungsprozesse verstanden werden würden.

Beiden DLT-Entwicklungsprojekten liegt die Erwartung zugrunde, die eigenen Bilddaten durch Automatisierung auszuwerten. Im ersten Fall geht es darum, die Analyse und ihre Güte zu verbessern sowie praktizierende Ärzt:innen im Klinikalltag zu entlasten und im zweiten Fall darum, die Analyse der Datenauswertung für die Erforschung überhaupt erst zu ermöglichen.

In beiden Fällen besitzen letztlich die Anwendenden die Daten z.B. in Form von Bildmaterial und sind in ihren Vorhaben darauf angewiesen, Programmierende einzubeziehen, um das jeweilige Ziel zu erreichen. Im Fallbeispiel A erscheint die DLT als Attraktionsobjekt, dessen allgemeine Versprechungen konkretisiert werden sollen, wobei es bei der letztlichen Umsetzung zu Schwierigkeiten kommt, sodass eine große Disbalance zwischen erster Aneignung und erfolgreicher Umsetzung sichtbar wird. DLT schafft hier also zunächst den Anschein leichter Umsetzbarkeit, der dann durch das Erfordernis für eine größere Datenbasis und disziplinäre Expertise konterkariert wird. Im Fallbeispiel B hingegen zeichnet sich sogleich der hohe Komplexitätsgrad des Problems ab, das bei den Biolog:innen zur Reaktion führt, disziplinäre Expertise zu beauf-

tragen, die dann die DLT zur Lösung einbringt und sie von den Ergebnissen profitieren lässt.

In beiden Fallbeispielen dieser anwendungsnahen Entwicklungsprojekte nehmen die Programmierenden die Funktion von Boundary Spannern ein. Ihre berufliche Ausbildung zeichnet sich dadurch aus, dass sie eine große Expertise in der Schnittmenge der Disziplinen aufbringen. Als Mediziningenieur und als Bioinformatiker besitzen beide schon qua Ausbildung Wissen über die Handlungs- und Wertlogiken von zwei Disziplinen. Sie bilden systematisch parallele Primärreferenzen – und somit eine doppelte Zugehörigkeit – aus, vermögen Felder zu überbrücken und sind so angeregt, Verständnis für die disziplinäre Logik und in diesem Fall das Erkenntnis- und Problemlösungsinteresse des Anwendungsfeldes zu entwickeln. Dennoch wird deutlich, dass die Entwicklung der Technologie und ihre Programmierung das Zentrum ihrer Expertise darstellen. Für B1 gilt dies noch mehr als für A1, da letzterer bereits durch seine Anstellung an einer universitären Klinik auch in praktischer Nähe zum Anwendungsfeld steht. Die Programmierenden sind in der Akteurskonstellation in der Position, die Grenze zwischen den Feldern überbrücken zu müssen. Aber an sie wird die Anforderung gerichtet, Probleme des Anwendungsfeldes zu lösen.

Zweite Beobachtung: Annotieren statt Verstehen

Generell gilt, dass qualitativ hochwertige und zahlenmäßig umfangreiche Lerndaten als »Goldstandard« (A2) betrachtet werden, um eine gute DLT zu entwickeln. Davon ausgehend zeigt sich, dass in den Projekten, in denen supervised DLT für automatisierte Bildanalyse entwickelt werden, die Phase der Datenannotierung⁶ ein konstitutives Element darstellt. Eine charakteristische Konstellation in den betrachteten Fallbeispielen tritt deutlich hervor: Bei der Annotierung – also der Aufbereitung des Datenmaterials als Lerndaten für den Algorithmus – ist die Einbindung der Anwendenden bzw. in unseren Fällen die Einbindung der Lebenswissenschaftler:innen existenziell.

In beiden betrachteten Fallbeispielen übernehmen die Anwendenden die Annotierung der Daten. Dies ist nötig, da die Annotierung der Daten nur mit

6 Im Folgenden verwenden wir bei den Ausführungen zu den Fallbeispielen statt des etablierten Begriffs »Annotation« den im Feld selbst genutzten Begriff der »Annotierung« und verbleiben somit in unmittelbarer Nähe zum Feld und (auch) seiner sprachlichen Praxis.

der spezifischen disziplinären Expertise vorgenommen werden kann. Es geht darum, fachlich fundierte Selektionsentscheidungen vorzunehmen und das Material so aufzubereiten, dass die DLT handhabbare Informationen daraus erhält.

In Fallbeispiel A steht dabei im Vordergrund, die radiologischen Aufnahmen mit den bestehenden schriftlichen Diagnosen aus Arztbriefen in skalierbare Daten umzuwandeln. So geht aus den schriftlichen Diagnosen nicht immer klar hervor, ob ein Aneurysma besteht oder nicht. Diese klare Unterscheidung wird aber seitens der DLT gefordert. Ausgehend von ihrer Expertise treffen praktizierende Radiolog:innen die Entscheidungen, wie diese ›Übersetzung‹ stattfinden kann und inwiefern schriftlich unklare und uneindeutige Diagnosen (z. B. »Blutung nicht sicher erkennbar« (A2)) in einen Score übertragen werden können. Die Aufbereitung der Lerndaten erfolgt somit in zwei Phasen. Zunächst gilt es, in einer Scanabbildung die entsprechenden Strukturen zu markieren, die als Aneurysma in Frage kommen. Bei der Detektion der Strukturen unterscheidet sich die Perspektive der DLT kaum von derjenigen der Ärzt:innen. Beide betrachten ein Bild und suchen nach ungewöhnlichen Mustern in den Daten. Deshalb können bestehende Diagnosen der vergangenen Jahre im Fall A einfach nochmal aufbereitet und gleichermaßen für den Learner übersetzt werden. Die darauffolgende Phase ist die der Klassifikation der detektierten Muster. Hierzu müssen die schriftlichen Diagnosen zu den entsprechenden detektierten Strukturen in einen Score umgewandelt werden.

Im Fallbeispiel B muss zunächst ein Tool programmiert werden, das die Daten zur Annotierung verfügbar macht. Mittels dieses von dem Programmierer entwickelten Tools können die Biolog:innen die relevanten Zellen in den einzelnen Bildern markieren und selektieren. »Genau, das ist so der erste Schritt, dass man einfach mal weiß, was will der Biologe überhaupt haben und dann dafür eben dann diese Label erstellt.« (B1) Dieses Tool ermöglicht zum einen, ein Verständnis des Programmierers für das Erkenntnisinteresse der Biolog:innen zu entwickeln, aber auch den Biolog:innen verständlich zu machen, was die Anforderungen für das Lernen und Programmieren der DLT sind. Die Expertise der Biolog:innen schlägt sich in diesem Prozess in der Annotierung der Lerndaten nieder und wird dann so aufbereitet, dass sie zugänglich für die DLT sind und in die Entwicklung eingespeist werden können. Dennoch besteht seitens des Programmierers keine genaue Kenntnis dessen, was die Label spezifisch bedeuten. So gilt es in dieser Projektphase eine Verständigung darüber zu erlangen, was das Erkenntnisinteresse der Anwendenden ist. Theore-

tisch übersetzt heißt dies, dass die Programmierenden hier den Schritt in das fremde Feld wagen müssen. Das Erkenntnisinteresse muss in die Technologie übersetzt werden, die jedoch selbst Anforderungen bereithält. Diese Anforderungen müssen dabei die Programmierenden vermitteln. In dieser Phase lässt sich die intensivste Zusammenarbeit erkennen.

Die Vorbereitung der Annotierung der Lerndaten findet in Kooperation statt und ist stark geprägt von einem jeweiligen »aufeinander Angewiesensein« auf die Expertise. Um die Technik zur Lösung des Problems nutzen zu können, bedarfes einer Verständigung darüber, wie die Daten aufbereitet werden müssen, sodass ein hochwertiger Lerndatenkorpus zustande kommt. Die Aufforderung zur Mitarbeit stellt aus Sicht des Programmierers B1 in dem Dienstleistungsverhältnis kein Problem dar: »Das ist quasi kein Problem, ihnen zu kommunizieren, dass wir jetzt irgendwie Annotationen von ihnen brauchen, um eben solche Modelle zu trainieren.« (B1)

In Fallbeispiel B fordert B1 als Programmierer die nötige Annotierung der Daten und ihrer Erfordernisse ein und schildert, was er braucht. Es findet aber in beiden Projekten eine Verständigung darüber statt, was die wechselseitigen Bedarfe sind. Die Kommunikation geht jedoch nicht über den Austausch der Bedarfe hinaus, sondern sie erfolgt vielmehr in Form einer »Übergabe« der Verantwortung für den darauffolgenden Teilschritt, der Durchführung der Annotierung. Die Verantwortlichkeiten sind nach wie vor verteilt und klar entlang der Projektphasen getrennt. Kommuniziert wird vor allem zur »Übergabe« in den nächsten Zuständigkeitsbereich, ohne dabei ganz genau zu verstehen, was das Gegenüber macht. Die beiden Disziplinen bleiben weitestgehend über die unterschiedlichen Zuständigkeiten und unterschiedlichen Expertisen, die eingebracht werden, voneinander getrennt und bringen durch die Verschiebungen wechselseitige Abhängigkeiten in den einzelnen Phasen hervor. »Im Prinzip bleibt einem nicht viel anderes übrig, genau. Also ich als Informatiker kann das teilweise nicht so richtig bewerten, was da jetzt genau die Annotierung vom Biologen ist.« (B1)

Die Programmierenden problematisieren diese Abhängigkeit vor allem vor dem Hintergrund, dass die Güte ihrer Arbeit (die DLT) von der Güte der Annotierung der Lerndaten abhängt, auf die sie nur bedingt Einfluss nehmen können, da es der spezifischen disziplinären Expertise bei der Aufbereitung bedarf. B1 problematisiert aus dieser Perspektive heraus, dass die Expert:innen bei der Annotierung der Daten auch müde werden und fehleranfällig sein können. Ob dies aber auftritt, kann er kaum kontrollieren. Aus seiner disziplinären Logik heraus entwickelt er deshalb schon Problemlösungen, die den

Einfluss der annotierenden Expert:innen verringern. Hiermit kann er diese existenzielle Projektphase selbst gestalten und die Kontrolle der Daten wieder übernehmen. B1 schildert, dass es verschiedene Möglichkeiten seitens der Programmierenden gibt, die manuelle, sehr aufwändige und zum Teil auch fehleranfällige Annotierung durch die Simulation und Modellierung künstlicher Trainingsdaten zu begegnen. Auch kann mit einer solchen Augmentierung die Lerndatenmenge künstlich erhöht werden. Hierbei werden die bestehenden Lerndaten simuliert und künstlich variiert. Hier kommt Künstliche Intelligenz zum Einsatz, um die händische Annotierung der Lerndaten zu ersetzen. B1 agiert hier nach der für das Feld der Programmierenden spezifischen Logik und macht das Problem zu seinem eigenen Forschungsfeld, um damit symbolisches Kapital in seinem Feld zu gewinnen.

Im Vergleich zu der vorherigen betrachteten Projektphase des Entstehungszusammenhangs zeigt sich in dieser Phase, dass es zu einem Wechsel in Abhängigkeit und Verantwortung kommt. Die Qualität der Datenannotierung ist ein ausschlaggebendes Erfolgskriterium für die DLT. In beiden Projekten wird der Relevanz der Qualität der Annotierung der Lerndaten begegnet und es werden verschiedene Methoden eingesetzt, damit die disziplinären Expert:innen aus den Anwendungsfeldern ihr Wissen in die Aufbereitung der Daten einbinden können. Dabei sind die Programmierenden auf die Güte der Annotierung angewiesen. Es zeigt sich in beiden Projekten, dass die Anwendenden nicht nur in den Entwicklungsprozess eingebunden werden, sondern sie als disziplinäre Expert:innen für diese Projekte existenziell sind. Doch darüber hinaus wird deutlich, welche große Probleme und Herausforderungen in dieser Phase bestehen. Die Logik der DLT fordert Daten ein, die Entscheidungen auf der Grundlage von Korrelationen ermöglichen. Umso mehr Daten vorhanden sind, mit denen die DLT trainiert werden kann, desto besser. Einzelne Fehler, die die Programmierenden in der Qualität nicht beurteilen können, können so durch die Masse kontrolliert werden. Doch beide Fälle zeigen, dass das Produzieren dieser Daten sehr viele Ressourcen in Anspruch nimmt und auch einen hohen Verständigungsaufwand erfordert. Während es im ersten Fall nicht über den Prototypen hinausgeht, entwickelt sich im zweiten Fall die Tendenz, die Expertise des Anwendungsfeldes durch weitere technische Applikationen zu ersetzen.

Dritte Beobachtung: DLT als Black-Box/Übergaben statt Kollaboration

DLT-Entwicklungsprojekte beinhalten weiterhin eine Phase der Programmierung bzw. eine Phase, in der der Algorithmus optimiert und kontrolliert wird. Häufig werden DLT als Blackbox beschrieben (siehe hierzu auch Beitrag Heckwolf/Schmitt in diesem Band). In dieser idealtypischen Projektphase geht es darum, dass die Programmierenden die Güte der DLT testen und diese kontinuierlich verbessern. B1 beschreibt, dass hierbei wichtig sei, »ein bisschen zu tricksen« und dieser Optimierungsprozess nach dem »Trial and Error«-Prinzip verlaufe. Dabei gäbe es durchaus »Patentrezepte«, die zur Verbesserung der Ergebnisse führen können. Dennoch kommt es hierbei »auf jeden Fall ein bisschen [auf] Erfahrung« an und darauf, »das Gespür zu haben, an welchen Stellschrauben« gedreht werden muss. An der hier beschriebenen Praxis der Programmierenden lassen sich die Übersetzungsschwierigkeiten gut erkennen. Sie entwickeln ein hohes Maß an implizitem praktischem Wissen, das sie bei der Anpassung des Learners einbringen. Nicht nur ist die DLT an sich schwer zu durchschauen – wie es technisch durch die *hidden layers* erwartet wird – sondern es ergibt sich auch aus dem praktischen Umgang mit ihm ein Problem der Verständigung über dessen Funktionsweise innerhalb der Projekte. Auch A1 beschreibt die Charakteristiken dieses Optimierungsvorgangs ähnlich:

»Bei mir war es tatsächlich etwas Glück. Ich habe an Parametern rumgespielt, die ich für sinnvoll erachtet habe, wo ich so das Gefühl hatte, ok, hier könnte man was dran schrauben und dann wird es besser.«

Von den Programmierenden selbst wird der Algorithmus als eine Blackbox und infolgedessen als nicht in Gänze kontrollierbar und verstehbar problematisiert. So bedarf es Erfahrung, Wissen und Glück, um die DLT zu optimieren. Dabei steht in den Beschreibungen der Befragten auch nicht das Verständnis der DLT im Fokus, sondern vor allem die Verbesserung der Güte der Ergebnisse.

»Das ist ja sehr, sehr kompliziert, posthoc rausfinden zu wollen, warum der Algorithmus so oder so gelernt hat. Da gibt es gewisse Verfahren, aber die sind so wahnsinnig aufwendig, die interessieren uns momentan auch nicht.«
(A2)

Der Fokus liegt auf dem Funktionieren der DLT und weniger auf dessen Verständnis. Die DLT gilt als funktionsfähig, wenn gewisse Fehlertoleranzen erfüllt werden. Die Programmierenden betonen, dass ihre Tätigkeiten vor allem anwendungsnah orientiert sind. Sie positionieren sich damit gegenüber theoretisch orientierten Informatiker:innen, deren Hauptaugenmerk dann auf einem durchdringenden Verständnis von DLT liege. Dies macht sichtbar, dass es für sie Schwierigkeiten bereitet, die Logiken der Technologie zu verstehen und entsprechend für Akteur:innen aus dem Anwendungsfeld zu übersetzen.

Die Akteur:innen aus dem Anwendungsfeld übergeben die Verantwortung in dieser Phase des Projektes in Gänze den Programmierenden. Die Gründe hierfür können unterschiedlich sein. So könnte es sein, dass sie nur an dem Outcome interessiert sind und den Programmierenden vertrauen oder sie glauben, weiterhin die Kontrolle darüber zu haben, wie die Technologie angewendet werden wird. Dies drückt insbesondere A2 aus, wenn er davon spricht, dass es letztlich die Ärzt:innen seien, die die Diagnose trafen und somit weiterhin die DLT kontrollieren. Im Folgenden »wechselt die Verantwortlichkeit wieder zurück zu den Ärzten in unserer Abteilung, die dann die gefundenen Ergebnisse auf Plausibilität überprüfen« (A2).

Doch die Abgabe der Verantwortung könnte auch auf ein tiefgreifendes Verständigungsproblem hindeuten. Vonseiten der Programmierenden wird die Verantwortungsabgabe seitens der Akteur:innen aus dem Anwendungsfeld kritisch hinterfragt. B1 gibt an, er nehme es so wahr, dass die Biolog:innen den Ergebnissen der Technik viel zu sehr vertrauen. Für B1 ist durch seine Erfahrungen mit der Programmierung von DLT klar, dass die Ergebnisse variieren können und stets auch Fehler auftauchen können. Scheinbar kann er dieses Verständnis den Biolog:innen nicht ausreichend vermitteln.

DLT als Blackbox wahrzunehmen führt in der Folge dazu, dass es zu einer stärkeren Trennung der beiden Handlungsfelder kommt. Die Optimierung der DLT läuft zwischen den Polen der Expertise der Programmierer:innen und deren Glück und Intuition entlang. Durch den Blackbox-Charakter von DLT entstehen spezifische Verständigungs- und Verstehenshürden und damit einhergehend unterschiedliche Deutungen der Akteurskonstellation. Dadurch, dass die DLT für die Beteiligten nicht voll durchsichtig und vermittelbar sind, kommt es zu Diskrepanzen in den Kontroll- und Expert:innenerwartung, die sich zwischen denen der Anwender:innen und der Programmierer:innen unterscheiden.

Die Wechsel in Zuständigkeit und Verantwortung werden von den Befragten dezidiert als solche beschrieben und im Prozess markiert. Dem stehen

Aussagen zur Zusammenarbeit gegenüber, die aufkommen, wenn es um die gesamte Projektkonstellation geht:

»Also es war auch schon so ein interdisziplinäres Projekt, wo man wirklich zusammen arbeiten musste aus zwei verschiedenen Disziplinen. Aus der Medizin, die nicht viel Ahnung von Technik, Programmieren und Deep-Learning an sich haben und ich, der halt mehr von der technischen Seite kommt, aber weniger Ahnung von Pathologien und solchen Sachen hat.«
(A1)

Das wechselseitige Aufeinander-angewiesen-sein und die Wechsel der Zuständigkeit samt der damit einhergehenden disziplinären Expertise erscheinen als charakteristische Elemente solcher anwendungsnahen DLT-Entwicklungsprojekte. Die Prozesse sind stetig von Übergaben der Verantwortung, Expertise und Kontrolle geprägt.

In der Analyse wird deutlich, dass es durch die DLT nicht zu einem weitreichenden und tiefergehenden Austausch über die disziplinären Verständnisse kommt. Es wird zwar kommuniziert, was das Verständnis der Projektpartner:innen ist, aber nur in Hinblick auf die Anforderungen des nächsten Prozessschritts. Es kommt dagegen vor allem zu Übergaben der Verantwortlichkeiten und weniger zu einer Verständigung über die jeweilige disziplinäre Logik. Die beschriebene existenzielle Einbindung der Anwendenden in den Prozess geschieht zwar und ist relevant, sie ist jedoch sehr reduktionistisch.

6. Diskussion

Die Forschung zu feldübergreifenden Entwicklungsprojekten hat gezeigt, dass die Entwicklungsgegenstände als Boundary Objects strukturieren können, welche Verhältnisse sich zwischen den beteiligten Akteur:innen entwickeln. Das betrifft auch das Verhältnis von Entwickler:innen und Anwender:innen. In unserem Beitrag haben wir argumentiert, dass sich DLT als Boundary Objects in Entwicklungsprojekten auf charakteristische Weise von anderen Technologien unterscheiden. Die explorative Analyse der beiden DLT-Entwicklungsprojekte hat drei Wirkweisen der DLT als Boundary Object offengelegt, die die Verhältnisse von DLT-Entwickler:innen und DLT-Anwender:innen auf spezifische Weise strukturieren.

Erstens zeigt unsere Analyse, dass DLT schon für die initiale Konstellation von DLT-Programmierenden und DLT-Anwendenden in einem Projektzusammenhang strukturierend sind. Als Technologie, die die Lösung von Problemen im möglichen Anwendungsfeld verspricht, gibt sie, wie viele andere Technologien auch, Anlass zu Feldgrenzen übergreifenden Entwicklungsprojekten. In der vorangehenden Forschung hat sich bereits gezeigt, dass die Tiefe der auf einen Entwicklungsgegenstand bezogenen Kenntnisse das Verhältnis grundlegend strukturiert (Leonardi/Bailey/Pierce 2019; Levina/Vaast 2005). In den beiden von uns betrachteten Fällen hat sich gezeigt, dass die DLT-Anwendenden sich selbst, trotz der teilweise beschriebenen leichten Zugänglichkeit der Technologie, besonders geringe auf die DLT bezogene Kenntnisse zuschreiben. Infolgedessen befassen sie sich im Entwicklungsprozess selbst nur rudimentär mit der Konstruktion der DLT und verstehen sich eher als Auftraggeber:innen denn als gleichberechtigte Entwicklungspartner:innen, während die Programmierenden als Dienstleister:innen wahrgenommen werden. Derartige Asymmetrien tauchen unabhängig vom Entwicklungsgegenstand auch in anderen Entwicklungszusammenhängen auf. Wir argumentieren aber, dass die DLT als Entwicklungsgegenstand eine Relationierung der Entwicklungspartner:innen als Auftraggeber:innen und Dienstleister:innen (gegenwärtig) aus zwei Gründen nahelegt. Erstens sind DLT im Gegensatz zu anderen Entwicklungstechnologien in vergleichsweise geringem Ausmaß in den (Anwendungs-)Feldern verbreitet und die potenziellen Anwendenden verfügen über vergleichsweise wenig technologiebezogenes Wissen. Zweitens sind DLT dadurch, dass sie selbst für die Programmierenden nur eingeschränkt zu durchschauen sind und in höherem Maße als Blackbox wahrgenommen werden, an sich besonders schwer verständlich. Die DLT-Anwendenden sind also aufgrund ihrer mangelnden DLT-Erfahrung und weil DLT an sich besonders schwer zu verstehen sind, kaum dazu befähigt, sich die Gestaltung der DLT gedanklich zu erschließen. Dass sich die DLT-Anwendenden im Entwicklungsprozess in eine eher passive Auftraggeber:innen-Rolle begeben, erscheint uns daher als typisch für (gegenwärtige) DLT-Entwicklungsprozesse.

Zweitens zeigt unsere Analyse, dass die DLT-Entwicklung spezifische DLT-Anwendungs-Einbindungspraktiken nahelegt und damit das Programmierungs-Anwendungs-Verhältnis grundlegend strukturiert. Um die DLT entwickeln zu können, sind die Entwickler:innen darauf angewiesen, dass die Anwender:innen ihnen adäquate Lerndaten zur Verfügung stellen. In den beiden betrachteten Fällen ist es dazu notwendig, dass die Anwender:innen auf

Grundlage ihrer Expertise Bilder annotieren. Anhand der Annotation identifizieren die DLT anschließend Muster, die es ihnen schließlich erlauben, Bilder nach eben diesen Mustern zu analysieren. Die Aufgabe der DLT-Entwickler:innen besteht im Wesentlichen darin, den Algorithmus für die spezifische Lernaufgabe zu sensibilisieren. Das Wissen der DLT-Anwendenden findet somit vergleichsweise unmittelbar Eingang in den Entwicklungsgegenstand. Damit ist die Art der Anwender:inneneinbindung im DLT-Entwicklungsprozess auf spezifische Weise vorstrukturiert und unterscheidet sich in der Art der Nutzer:inneneinbindung systematisch von anderen Technikentwicklungsprozessen. Einerseits fordert die DLT die Einbindung von Nutzenden: Während die Entwickler:innen von Maschinen oder klassischer Software in vielen Fällen auch ohne die Einbindung von Anwendenden, alleine ausgehend von ihren eigenen Vorstellungen von Anwendungsfällen, operieren, ist die frühe Einbindung von Anwendenden typisch für die DLT-Entwicklung. Andererseits bleibt die Einbindung aber typischerweise oberflächlich und maschinell. In vielen Technikentwicklungsprozessen wird die Einbindung von Anwendenden durch aufwändige Wissenstransferpraktiken realisiert. Entwickler:innen folgen den Anwender:innen dazu mitunter tagelang durch ihren Alltag, um auch ihr implizites Wissen und dessen kontextuelle Einbettung zu erfassen (Nonaka/Takeuchi 1995; Berends et al. 2006). Entwickler:innen arbeiten hier dementsprechend intensiv daran, Anwender:innen zu verstehen und sich ihr Anwender:innenwissen anzueignen. Die Feldgrenzen, die durch die Differenzen im praktischen Wissen der Akteur:innen definiert sind, verschwimmen dadurch im Erfolgsfall temporär (Leonardi/Bailey/Pierce 2019; Carlile 2002). Dagegen sind die DLT-Entwickler:innen lediglich darauf angewiesen, zu verstehen, welche Mittel sie zur Annotation benötigen. Ein Verstehen der Anwender:innen im engeren und tieferen Sinne ist dabei nicht erforderlich, wenn auch der alltägliche Anwendungskontext weitestgehend ausgeblendet bleibt. Zwar steht es auch den DLT-Entwickler:innen prinzipiell offen, sich intensiver mit dem Wissen der Anwender:innen zu befassen. Die DLT legt das aber zumindest nicht nahe und es erscheint uns daher sehr wahrscheinlich, dass es in DLT-Entwicklungsprojekten typischerweise bei der oberflächlichen und maschinenvermittelten Anwender:inneneinbindung bleibt.

Drittens führen die Undurchsichtigkeit der DLT (Blackbox) und die Möglichkeit, das erforderliche Nutzer:innenwissen maschinell zu integrieren, gemeinsam dazu, dass DLT-Entwickler:innen und DLT-Anwender:innen in den betrachteten DLT-Entwicklungsprojekten eher nebeneinander und unabhän-

gig voneinander als gemeinsam arbeiten (Chettiparamb 2007: 16). Auch in Folge der Arbeitsorganisationen bleiben die beiden Felder daher unabhängig und eigenständig. Eine partielle oder temporäre Integration der beteiligten Felder, wie sie durch andere feldübergreifende Entwicklungsprojekte durchaus vorangetrieben wird, wird in den DLT-Entwicklungsprojekten daher kaum gefördert. Die DLT als zentrales Boundary Object trägt in den betrachteten Entwicklungsprojekten also eher dazu bei, dass die Grenzen zwischen den Feldern trotz Kollaboration aufrechterhalten werden können, als dass sie deren Überwindung befördert.

7. Konklusion

Mit der Analyse der Wirkung von DLT als Boundary Object in Entwicklungsprojekten leistet diese Arbeit insbesondere zwei Beiträge. Mit unserer kritischen Analyse der Wirkung von Boundary Objects tragen wir zu einem differenzierten Verständnis des Konzepts bei. Neben der bereits benannten Machtasymmetrie stiftenden Wirkung weist unser Beitrag darauf hin, dass Boundary Objects nicht nur dabei helfen können, Grenzen zu überwinden, sondern es ebenso möglich ist, dass sie das Aufrechterhalten dieser Grenzen trotz Kollaboration fördern. Am Beispiel von DLT haben wir bereits einige Charakteristika von Boundary Objects identifiziert, die eine derartige Wirkung begünstigen. Es scheint aber äußerst fruchtbar in weiterführender Forschung andere Typen von Boundary Objects miteinander zu vergleichen und detaillierter herauszuarbeiten, welche Boundary Objects-Qualitäten ausschlaggebend dafür sind, ob Boundary Objects eher zur Überwindung oder zur Stabilisierung von Grenzen in Kollaborationsprojekten beitragen.

Zweitens adressiert unser Beitrag die Forschung zu KI und DLT. Viele diesbezügliche Beiträge beschäftigen sich mit den möglichen Auswirkungen der Anwendung von KI in unterschiedlichen Bereichen. Die Techniksoziologie hat jedoch immer wieder gezeigt, dass Bilder von Technik und technische Möglichkeiten bis zu ihrer Anwendung im Alltag einen vielschichtigen sozialen Gestaltungsprozess durchlaufen, der maßgeblich beeinflusst, wie eine vorhandene Technologie tatsächlich wirksam wird. Es erscheint daher sinnvoll, sich diesem Gestaltungsprozess selbst stärker zuzuwenden, um die rezenten Entwicklungen erschließen zu können. Wir haben dieses Ziel in diesem Beitrag mit Blick auf DLT-Entwicklungsprojekte verfolgt. Unsere Befunde weisen darauf hin, dass DLT-Entwicklungsprojekte Anwender:innen und Entwick-

ler:innen in ein spezifisches Verhältnis zueinander bringen. Unsere Daten belegen, dass die beiden Felder im Vergleich zu anderen Entwicklungs- und Kollaborationsprojekten in hohem Maße unabhängig voneinander bleiben und ein wechselseitiges Verstehen nur an der Oberfläche stattfindet. Mit Blick auf die bisherige Forschung zu Akteurskonstellationen und den Ergebnissen von Entwicklungsprojekten erscheinen daran anschließend zwei Szenarien plausibel. Erstens könnte die mangelhafte Anwender:innen-Integration dazu führen, dass die ›inselbegabten‹ Anwendungen, die aus den Projekten hervorgehen, nicht in ausreichendem Maße mit den alltäglichen Anforderungen der Anwender:innen kompatibel sind und daher oftmals nicht dort etabliert werden können. Zweitens führt die Unabhängigkeit der Partner:innen im Entwicklungsprozess dazu, dass die Entwickler:innen die Gestaltung der Anwendungen dominieren. Die Forschung zur Entwicklung neuer Produkte zeigt, dass Entwickler:innen dazu neigen, unhinterfragte Annahmen aus ihrer eigenen sozialen Lebenswelt in die Produkte einzuschreiben. Indem Anwender:innen diesen Regeln entsprechen müssen, wenn sie die Produkte nutzen wollen, werden die Logiken des Entwicklungsfeldes durch die Technik im Anwendungsfeld gewissermaßen reproduziert. Die Eigenständigkeit der DLT-Entwickler:innen führt dazu, dass sie die Logik ihres Feldes weitestgehend ungefiltert in die DLT-Anwendungen einschreiben können. Wenn sich die entsprechenden DLT im Anwendungsfeld etablieren, ist dementsprechend eine Angleichung an das Feld der Programmierer:innen zu erwarten. Weiterführende Forschung, die die konkreten Konsequenzen des Verhältnisses von Anwender:innen und Entwickler:innen für die Ausgestaltung der entwickelten Anwendungen empirisch untersucht, erscheint daher äußerst relevant.

An dieser Stelle nun lässt sich die Perspektive Whites sinnvoll einbinden, indem sie eine andere Perspektive auf das Phänomen ermöglicht. Die hier angewendete feldtheoretische Perspektive folgt dem Impuls, zu klären, warum verinnerlichte Prägungen, die in sozialen Feldern von Akteur:innen gewonnen wurden, ihre Persistenz in divergierenden Kontexten erhalten. Der oder die Akteur:in erhält damit eine relativ stabile sozialräumliche Verortung und wird als zugehörig zu einem bestimmten Feld klassifiziert (Hirschauer 2017: 30ff.). Die Beobachtung von Diskontinuitäten zwischen Feldern ist damit in gewisser Hinsicht vorprogrammiert.⁷

7 Diese eher klassische Lesart der bourdieuschen Feldtheorie wurde zuletzt durch Schmitz et al. (2017) in Frage gestellt. Sie verweisen, um die Grenzen von Feldern zu bestimmen, darauf, sich die Interdependenzen zwischen Feldern genauer anzusehen.

Im Anschluss an White kann allerdings grundsätzlich davon ausgegangen werden, dass Akteur:innen stets in unterschiedlichen Netzwerken eingebettet sind. Dementsprechend stellt sich hier eher die Frage, ob und wie Akteur:innen Kompetenzen darin erlangen, in unterschiedlichen Beziehungen mit heterogenen Handlungsregeln erfolgreich handeln zu können. Der Wechsel zwischen den Kontexten wird bei Whites Einführung des *netdom* Begriffes deutlich: »Identities switch from netdom to netdom, finding footings in different networks in differing domain contexts« (White 2008: 7)⁸. Resultat und Ziel dieser Theorie ist keine saubere und geordnete Verortung des Sozialraums, wie sie die Theorie der sozialen Felder in metaphorischer Anlehnung an die physikalischen Kraftlinien relativ dazu leistet, denn sie geht von einer grundsätzlichen Unordnung der Welt aus (vgl. White 2008: 18). Dies bedeutet in Hinblick auf den zugrunde liegenden Forschungsgegenstand der Zusammenarbeit in Entwicklungsprojekten mit DLT, dass es mittels der Perspektive Whites interessant sein könnte, die tatsächlichen Kommunikationsprozesse zwischen den Partner:innen beobachtend zu folgen und nicht nur die Darstellung in den Interviews zu interpretieren, um genauer herauszuarbeiten, ob sich in der Theoriesprache Whites durch die Events und spezifischen Netzwerksituationen der Entwicklungsprojekte Identitäten und Netzwerkdomänen ausbilden, die Kontrolle ausüben. Es müsste darum gehen, die Domänen herauszuarbeiten, die Mische/White (1998: 702) wie folgt definieren:

So erst werde die relative Autonomie oder Heteronomie eines Feldes zwischen anderen Feldern sichtbar, sodass sich eine stabile sozialräumliche Verortung auch erst in Feldern mit hoher Autonomie ergebe.

- 8 Die Art des Wechsels bzw. Überlagerung und was White dabei unter Identitäten versteht, definieren Mohr/White (2008: 489) folgendermaßen: »White (1992, 2008) provides extensive theoretical specification that begins not with individual embodied persons, but rather with identities that can be generalized to include any source of action. A firm, a community, a crowd, oneself on the tennis court, encounters of strangers on a sidewalk, each may be identities. Identities are triggered out of events—which is to say, relational situations that emerge from the flow of social events, switching from one network situation into another, that produce a need for control over uncertainty and thus a control over situated actions and fellow identities. Lives are organized around this flowing movement from one network situation into another and these situations are themselves phenomenological sites that include certain other identities (in particular networks) organized around specific domains of activity (hence they are network/domains or simply netdoms).«

»We define domain as the perceived array of such signals – including story sets, symbols, idioms, registers, grammatical paternings, and accompanying soporeal markers – that characterize a particular specialized field of interaction.«

Innerhalb der Kommunikationsprozesse wäre dann nachzuvollziehen, ob es zu ›Situationen‹ kommt also »those problematic, high-stakes episodes that cast our prescribed roles and trajectories into question [...] each involves a previously predictable, stylized interaction that suddenly becomes fraught with uncertainty, danger, and/or opportunity« (ebd.: 698). Es ginge darum, die tatsächlichen Vernetzungen nachzuvollziehen, den Wechseln der Identitäten zwischen den jeweiligen Kontexten zu folgen und die Geschichten innerhalb des Netzwerks herauszuarbeiten:

»Active participation (control efforts) in these situations depends upon both the network logic and the shared conceptual understanding of what is going on in the netdom. The shared understanding (localized meaning) is produced through the interactions (and especially the switching back and forth across these interactional sites) and embodied in a shared repertoire of netdom stories that provide the basis for a common understanding of what is going on in the moment.« (Mohr/White 2008: 489)

Dies ermöglicht es weiter zu hinterfragen, inwiefern es in den spezifischen Entwicklungsprojekten gelingt, ein gemeinsames Verständnis aufzubauen oder ob letztlich weiterhin Wege gesucht werden, den Verständigungsprozess zu umgehen und den Herausforderungen, die ein solcher bereitstellt, auszuweichen.

8. Literatur

Amershi, Saleema/Begel, Andrew/Bird, Christian/DeLine, Robert/Gall, Harald/Kamar, Ece/Nagappan, Nachiappan/Nushi, Besmira/Zimmermann, Thomas (2019): »Software Engineering for Machine Learning: A Case Study«, in: 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), IEEE, S. 291–300.

- Aust, Holger (2021): »KI: Hype oder Technologie der Zukunft?«, in: Holger Aust (Hg.), *Das Zeitalter der Daten*, Berlin/Heidelberg: Springer Berlin Heidelberg, S. 21–32.
- Barley, Stephen R./Tolbert, Pamela S. (1997): »Institutionalization and Structuration: Studying the Links between Action and Institution«, in: *Organization Studies* 18, S. 93–117.
- Barrett, Michael/Oborn, Eivor (2010): »Boundary object use in cross-cultural software development teams«, in: *Human Relations* 63, S. 1199–1221.
- Battilana, Julie (2006): »Agency and Institutions. The Enabling Role of Individuals' Social Position«, in: *Organization* 13, S. 653–676.
- Bechky, Beth A. (2003): »Sharing Meaning Across Occupational Communities. The Transformation of Understanding on a Production Floor«, in: *Organization Science* 14, S. 312–330.
- Berends, Hans/Bij, Hans/Debackere, Koenraad/Weggeman, Mathieu (2006): »Knowledge sharing mechanisms in industrial research«, in: *R and D Management* 36, S. 85–95.
- Berends, Hans/Sydow, Jörg (2019): »Introduction: Process Views on Inter-organizational Collaborations«, in: Hans Berends/Jörg Sydow (Hg.), *Managing Inter-organizational Collaborations: Process Views*, Bingley: Emerald Publishing Limited, S. 1–10.
- Berger, Peter L./Luckmann, Thomas (1980): *Die gesellschaftliche Konstruktion der Wirklichkeit. Eine Theorie der Wissenssoziologie*, Frankfurt a.M.: Fischer Taschenbuch.
- Bjerregaard, Toke (2010): »Industry and academia in convergence: Micro-institutional dimensions of R&D collaboration«, in: *Technovation* 30, S. 100–108.
- Bødker, Susanne (1998): »Understanding Representation in Design«, in: *Human–Computer Interaction* 13, S. 107–125.
- Bødker, Susanne (2000): »Scenarios in user-centred design-setting the stage for reflection and action«, in: *Interacting with Computers* 13, S. 61–75.
- Bogers, Marcel/Zobel, Ann-Kristin/Afuah, Allan/Almirall, Esteve/Brunswick, Sabine/Dahlander, Linus/Frederiksen, Lars/Gawer, Annabelle/Gruber, Marc/Haeffliger, Stefan/Hagedoorn, John/Hilgers, Dennis/Laursen, Keld/Magnusson, Mats G./Majchrzak, Ann/McCarthy, Ian P./Moeslein, Kathrin M./Nambisan, Satish/Piller, Frank T./Radziwon, Agnieszka/Rossi-Lamastra, Cristina/Sims, Jonathan/Ter Wal, Anne L. J. (2017): »The open innovation research landscape: established perspectives

- and emerging themes across different levels of analysis«, in: *Industry and Innovation* 24, S. 8–40.
- Borchardt, Andreas/Göthlich, Stephan E. (2007): »Erkenntnisgewinnung durch Fallstudien«, in: Sönke Albers/Daniel Klapper/Udo Konradt/Achim Walter/Joachim Wolf (Hg.), *Methodik der empirischen Forschung*, Wiesbaden: Gabler, S. 33–48.
- Bourdieu, Pierre (1993): *Sozialer Sinn. Kritik der theoretischen Vernunft* (= Suhrkamp-Taschenbuch Wissenschaft, Band 1066), Frankfurt a.M.: Suhrkamp.
- Bourdieu, Pierre (2009): *Entwurf einer Theorie der Praxis auf der ethnologischen Grundlage der kabyllischen Gesellschaft* (= Suhrkamp-Taschenbuch Wissenschaft, Band 291), Frankfurt a.M.: Suhrkamp.
- Bourdieu, Pierre/Wacquant, Loïc J. D. (2006): *Reflexive Anthropologie* (= Suhrkamp Taschenbuch Wissenschaft, Band 1793), Frankfurt a.M.: Suhrkamp.
- Buchkremer, Rüdiger/Witte, Bart de/Matusiewicz, David (2020): »KI in Gesundheit und Medizin«, in: Rüdiger Buchkremer/Thomas Heupel/Oliver Koch (Hg.), *Künstliche Intelligenz in Wirtschaft & Gesellschaft*, Wiesbaden: Springer Fachmedien Wiesbaden, S. 385–395.
- Carlile, Paul R. (2002): »A Pragmatic View of Knowledge and Boundaries: Boundary Objects in New Product Development«, in: *Organization Science*, S. 442–455.
- Chettiparamb, Angélique (2007): *Interdisciplinarity: a literature review*, Southampton: University of Southampton.
- Degele, Nina (1996): »Die Entwicklung und Nutzung von Software. Zur Genese informationstechnischen Handelns«, in: *Zeitschrift für Soziologie* 25, S. 58–70.
- Fligstein, Neil/McAdam, Doug (2011): »Toward a General Theory of Strategic Action Fields«, in: *Sociological Theory* 29, S. 1–26.
- Friedland, Roger/Alford, Robert R. (1991): »Bringing Society Back In: Symbols, Practices, and Institutional Contradictions«, in: Walter W. Powell/Paul DiMaggio (Hg.), *The New institutionalism in organizational analysis*, Chicago: University of Chicago Press, S. 232–266.
- Gassol, Jeaninne H. (2007): »The effect of university culture and stakeholders' perceptions on university–business linking activities«, in: *The Journal of Technology Transfer* 32, S. 489–507.
- Giddens, Anthony (1997): *Die Konstitution der Gesellschaft. Grundzüge einer Theorie der Strukturierung*, Frankfurt/New York: Campus.

- Gläser, Jochen (2006): *Wissenschaftliche Produktionsgemeinschaften. Die soziale Ordnung der Forschung* (= Campus Forschung, Bd. 906), Frankfurt a.M./New York: Campus-Verlag.
- Hahn, Katrin (2013): *Heterogene Akteure als Innovationspartner. Zur Strukturierung von Handeln in industriellen Innovationsprojekten*, Wiesbaden: Springer VS.
- Hallett, Tim (2003): »Symbolic Power and Organizational Culture«, in: *Sociological Theory* 21, S. 128–149.
- Hawkins, Beverley/Pye, Annie/Correia, Fernando (2017): »Boundary objects, power, and learning: The matter of developing sustainable practice in organizations«, in: *Management Learning* 48, S. 292–310.
- Heidenreich, Martin/Mattes, Jannika (2017): »Kollaborative Innovationen. Die innerbetriebliche Nutzung externer Wissensbestände in vernetzten Entwicklungsprozessen«, in: Martin Heidenreich/Jürgen Kädtler/Jannika Mattes (Hg.), *Kollaborative Innovationen. Die innerbetriebliche Nutzung externer Wissensbestände in vernetzten Entwicklungsprozessen*, Göttingen: Göttingen University Press, S. 13–43.
- Hippel, Eric von (1986): »Lead Users: A Source of Novel Product Concepts«, in: *Management Science* 32, S. 791–805.
- Hirschauer, Stefan (Hg.) (2017): *Un/doing Differences. Praktiken der Human-differenzierung*, Weilerswist: Velbrück Wissenschaft.
- Hosny, Ahmed/Parmar, Chintan/Quackenbush, John/Schwartz, Lawrence H./Aerts, Hugo J. W. L. (2018): »Artificial intelligence in radiology«, in: *Nature reviews. Cancer* 18, S. 500–510.
- Ibert, Oliver/Müller, Felix C./Stein, Axel (2014): *Produktive Differenzen. Eine dynamische Netzwerkanalyse von Innovationsprozessen* (= Science Studies), Bielefeld: transcript.
- Knorr Cetina, Karin (2002): *Wissenskulturen. Ein Vergleich naturwissenschaftlicher Wissensformen* (= Suhrkamp-Taschenbuch Wissenschaft, Band 1594), Frankfurt a.M.: Suhrkamp.
- Krais, Beate (2004): »Habitue und soziale Praxis«, in: Magareta Steinrück (Hg.), *Pierre Bourdieu, Politisches Forschen, Denken und Eingreifen*, Hamburg: VSA, S. 91–106.
- Kuckartz, Udo (2018): *Qualitative Inhaltsanalyse. Methoden, Praxis, Computerunterstützung* (= Grundlagentexte Methoden), Weinheim: Beltz.
- Lamnek, Siegfried (2008): *Qualitative Sozialforschung. Lehrbuch* (= Beltz PVU Lehrbuch), Weinheim, Basel: Beltz PVU.

- Leonardi, Paul M./Bailey, Diane E./Pierce, Casey S. (2019): »The Coevolution of Objects and Boundaries over Time: Materiality, Affordances, and Boundary Salience«, in: *Information Systems Research* 30, S. 665–686.
- Levina, Natalia/Vaast, Emmanuelle (2005): »The Emergence of Boundary Spanning Competence in Practice: Implications for Implementation and Use of Information Systems«, in: *MIS Quarterly* 29, S. 335.
- Levina, Natalia/Vaast, Emmanuelle (2008): »Innovating or doing as Told? Status Differences and Overlapping Boundaries in Offshore Collaboration«, in: *MIS Quarterly* 32, S. 307.
- Lizardo, Omar (2004): »The Cognitive Origins of Bourdieu's Habitus«, in: *Journal for the Theory of Social Behaviour* 34, S. 375–401.
- Majchrzak, Ann/Jarvenpaa, Sirkka L./Bagherzadeh, Mehdi (2015): »A Review of Interorganizational Collaboration Dynamics«, in: *Journal of Management* 41, S. 1338–1360.
- Martin, John L. (2003): »What Is Field Theory?«, in: *American Journal of Sociology* 109, S. 1–49.
- Mische, Ann/White, Harrison C. (1998): »Between Conversation and Situation: Public Switching Dynamics across Network Domains«, in: *Social Research* 65, S. 695–724.
- Mohr, John W./White, Harrison C. (2008): »How to model an institution«, in: *Theory and Society* 37, S. 485–512.
- Nonaka, Ikujiro/Takeuchi, Hirotaka (1995): *The knowledge-creating company*, New York/Oxford: Oxford University Press.
- Razzak, Muhammad I./Naz, Saeeda/Zuib, Ahmad (2018): »Deep Learning for Medical Image Processing: Overview, Challenges and the Future«, in: Nilanjan Dey/Amira S. Ashour/Surekha Borra (Hg.), *Classification in BioApps*, Cham: Springer International Publishing, S. 323–350.
- Roth, Philip (2019): »Die feldspezifische Eigenlogik der praktischen Konstitution sozialer Netzwerke. Konzeptionelle Überlegungen und beispielhafte empirische Befunde«, in: Jan Fuhse/Karoline Krenn (Hg.), *Netzwerke in gesellschaftlichen Feldern*, Wiesbaden: Springer Fachmedien Wiesbaden, S. 171–198.
- Roth, Philip/Diefenbach, Nadine (2021): »The Constitution of Boundaries: How the Embeddedness of Organizational Users Structures the Transfer of their Knowledge«, in: *NOvation – Critical Studies of Innovation* 3, S. 134–162.
- Schmitt, Marco (2019): »Felder und Netzwerkdomänen in der Wissenschaft«, in: Jan Fuhse/Karoline Krenn (Hg.), *Netzwerke in gesellschaftlichen Feldern*, Wiesbaden: Springer Fachmedien Wiesbaden, S. 63–79.

- Schmitz, Andreas/Witte, Daniel/Gengnagel, Vincent (2017): »Pluralizing field analysis: Toward a relational understanding of the field of power«, in: *Social Science Information* 56, S. 49–73.
- Serre, Thomas (2019): »Deep Learning: The Good, the Bad, and the Ugly«, in: *Annual review of vision science* 5, S. 399–426.
- Star, Susan L./Griesemer, James R. (1989): »Institutional ecology, translations and boundary objects: Amateurs and professionals in Berkeley's Museum of Vertebrate Zoology, 1907–39«, in: *Social Studies of Science* 19, S. 387–420.
- White, Harrison C. (2008): *Identity and Control. How Social Formations Emerge*, Princeton: Princeton University Press.
- Yin, Robert K. (2003): *Case study research. Design and methods* (= Applied social research methods series, Band 5), Thousand Oaks, California: Sage.

